

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method ~~for adjusting the color information of an image by modelling a non-linear transfer function with a power law function, the method~~ comprising:

receiving a transfer function, wherein the transfer function specifies a set of output values corresponding to a set of input values;

iteratively, until a termination flag is set:

receiving a first power law function;

generating an auxiliary function ~~from the~~ including modifying the transfer function ~~and~~ using local differences between the transfer function and the first power law function;

~~fitting a second power law function to the auxiliary function~~ defining a second power law function by fitting a power law function to the auxiliary function;

calculating a modelling error ~~from~~ between the second power law function and the transfer function, wherein the modelling error is an error value obtained from ~~a plurality of~~ local differences between the second power law function and the transfer function; ~~and~~

providing the second power law function as the first power law function for the next iteration when the modelling error is greater than or equal to a predetermined value; and

setting the termination flag when the modelling error is less than [[a]] the predetermined value; and

using the second power law function to calculate a gamma value that is used to adjust the color information of the image,

where in a first iteration, the first power law function is defined by fitting a power law function to the transfer function.

2. (Currently Amended) The method of claim 1, where modifying the transfer function using local differences includes determining a difference for one or more input values between the transfer function and the first power law function. ~~3, wherein in an iteration other than the first iteration, the first power law function is the second power law function that was fit to the auxiliary function in the immediately preceding iteration.~~

3. (Currently Amended) The method of claim 1, where modifying the transfer function using local differences includes determining one or more closest points between the transfer function and the first power law function without regard for the input value associated with the respective points. ~~wherein in a first iteration, the first power law function is a power law function obtained by fitting the transfer function.~~

4. (Original) The method of claim 1, further comprising:

counting the number of iterations; and

setting the termination flag when the number of iterations exceeds a maximum number of iterations.

5. (Original) The method of claim 1, wherein:

the transfer function is a transfer function for gamma correction, and the first and second power law functions are power law functions having a form of cx^{β}, wherein x is the input variable of the power law functions, and c and β are real numbers.

6. (Original) The method of claim 5, wherein:

fitting the second power law function to the auxiliary function includes fitting a linear function to a logarithmic representation of the auxiliary function.

7. (Original) The method of claim 6, wherein:

fitting the linear function to the logarithmic representation of the auxiliary function includes minimizing a least square error between the linear function and the logarithmic representation of the auxiliary function.

8. (Previously Presented) The method of claim 1, further comprising:

using a modifying parameter to weight the local differences between the transfer function and the first power law function, and using the weighted local differences to generate the auxiliary function from the transfer function.

9. (Cancelled)

10. (Currently Amended) The method of claim 8, [[,]]further comprising:

generating a plurality of second auxiliary functions from the transfer function, the first power law function and a corresponding plurality of modifying parameters, wherein each auxiliary function is generated by weighting the local differences between the transfer function and the first power law function using a corresponding one of the plurality of modifying parameters;

fitting each of the plurality of second auxiliary functions to generate a respective plurality of second power law functions, where each of the plurality of second power law functions corresponds to one of the plurality of modifying parameters;

calculating a plurality of modelling errors between the transfer function and each of the plurality of second power law functions, wherein each of the modeling errors corresponds to one of the plurality of modifying parameters;

executing a minimization procedure to determine a value of ~~the~~ a particular modifying parameter of the plurality of modifying parameters that minimizes the modelling error, ~~and~~

using the value of the particular modifying parameter that minimizes the modelling error to weight the local differences between the transfer function and the first power law function; and

generating the auxiliary function using the particular modifying parameter, the transfer function, and the first power law function.

11. (Previously Presented) The method of claim 10, wherein executing the minimization procedure comprises fitting a quadratic function to a distribution of modelling errors as a function of the plurality of modifying parameters.

12. (Previously Presented) The method of claim 10, wherein executing the minimization procedure comprises executing a golden search algorithm.

13. (Original) The method of claim 1, wherein: calculating the modelling error for the second power law function comprises calculating a total square error between the transfer function and the second power law function.

14. (Original) The method of claim 1, wherein: calculating the modelling error for the second power law function comprises calculating the maximum absolute difference between the transfer function and the second power law function.

15. (Original) The method of claim 1, wherein: receiving a transfer function comprises receiving a plurality of transfer function values.

16. (Previously Presented) The method of claim 1, wherein: receiving a transfer function comprises receiving a piecewise continuous monotonically increasing function.

17. (Currently Amended) A computer-implemented method ~~for adjusting the color information of an image by modelling a non-linear transfer function with a power law function, the method~~ comprising:

receiving a transfer function, wherein the transfer function specifies a set of output values corresponding to a set of input values;

fitting the transfer function with a first power law function;

iteratively, until a termination flag is set:

reflecting the first power law function about the transfer function to generate an auxiliary function;

~~fitting the auxiliary function with a second power law function~~ defining a second power law function by fitting a power law function to the auxiliary function;

calculating a modelling error ~~from~~ between the second power law function and the transfer function, wherein the modelling error is an error value obtained from ~~a plurality of~~ local differences between the second power law function and the transfer function;

identifying the first power law function with the second power law function for a

next iteration when the modelling error is greater than or equal to a predetermined value; and

setting the termination flag when the modelling error is less than [[a]] the predetermined value; and

using the second power law function to calculate a gamma value that is used to adjust the color information of the image.

18. (Currently Amended) A computer program product, implemented on a machine readable storage device, ~~for adjusting the color information of an image by modelling a non-linear transfer function with a power law function,~~ the computer program product comprising instructions operable to cause a programmable processor to:

receive a transfer function, wherein the transfer function specifies a set of output values corresponding to a set of input values;

iteratively, until a termination flag is set:

receive a first power law function;

generate an auxiliary function ~~from the~~ including modifying the transfer function ~~and~~ using local differences between the transfer function and the first power law function;

~~fit a second power law function to the auxiliary function~~ define a second power law function by fitting a power law function to the auxiliary function;

calculate a modelling error ~~from~~ between the second power law function and the transfer function, wherein the modelling error is an error value obtained from ~~a plurality of~~ local differences between the second power law function and the transfer function; ~~and~~

provide the second power law function as the first power law function for the next iteration when the modelling error is greater than or equal to a predetermined value; and

set the termination flag when the modelling error is less than [[a]] the predetermined value; and

use the second power law function to calculate a gamma value that is used to adjust the color information of the image,

where in a first iteration, the first power law function is defined by fitting a power law function to the transfer function.

19. (Currently Amended) The computer program product of claim 18, where modifying the transfer function using local differences includes determining a difference for one or more input values between the transfer function and the first power law function.~~20, wherein in an iteration other than the first iteration, the first power law function is the second power law function that was fit to the auxiliary function in the immediately preceding iteration.~~

20. (Currently Amended) The computer program product of claim 18, where modifying the transfer function using local differences includes determining one or more closest points between the transfer function and the first power law function without regard for the input value associated with the respective points. ~~wherein in a first iteration, the first power law function is obtained by fitting the transfer function.~~

21. (Previously Presented) The computer program product of claim 18, further comprising instructions operable to cause the programmable processor to count the number of iterations; and to

set the termination flag when the number of iterations exceeds a maximum number of iterations.

22. (Previously Presented) The computer program product of claim 18, wherein:

the transfer function is a transfer function for gamma correction, and the first and second power law functions are power law functions having a form of cx^{β}, wherein x is the input variable of the power law functions, and c and β are real numbers.

23. (Previously Presented) The computer program product of claim 22, wherein the instructions to fit the second power law function to the auxiliary function includes instructions to fit a linear function to a logarithmic representation of the auxiliary function.

24. (Previously Presented) The computer program product of claim 23, wherein the instructions to fit the linear function to the logarithmic representation of the auxiliary function includes instructions to minimize a least square error between the linear function and the logarithmic representation of the auxiliary function.

25. (Previously Presented) The computer program product of claim 18, further comprising instructions operable to cause the programmable processor to use a modifying parameter to weight the local differences between the transfer function and the first power law function, and to use the weighted local differences to generate the auxiliary function from the transfer function.

26. (Cancelled)

27. (Currently Amended) The computer program product of claim 25, further comprising instructions to:

generate a plurality of second auxiliary functions from the transfer function, the first power law function and a corresponding plurality of modifying parameters, wherein each auxiliary function is generated by weighting the local differences between the transfer function and the first power law function using a corresponding one of the plurality of modifying parameters;

fit each of the plurality of second auxiliary functions to generate a respective plurality of second power law functions, where each of the plurality of second power law functions corresponds to one of the plurality of modifying parameters;

calculate a plurality of modelling errors between the transfer function and each of the plurality of second power law functions, wherein each of the modelling errors corresponds to one of the plurality of modifying parameters;

execute a minimization procedure to determine a value of ~~the~~ a particular modifying parameter of the plurality of modifying parameters that minimizes the modelling error; and

use the value of the particular modifying parameter that minimizes the modelling error to weight the local differences between the transfer function and the first power law function; and

generate the auxiliary function using the particular modifying parameter, the transfer function, and the first power law function.

28. (Previously Presented) The computer program product of claim 27, wherein the instructions to execute the minimization procedure comprise instructions to fit a quadratic function to a distribution of modelling errors as a function of the plurality of modifying parameters.

29. (Previously Presented) The computer program product of claim 27, wherein the instructions to execute the minimization procedure comprise instructions to execute a golden search algorithm.

30. (Previously Presented) The computer program product of claim 18, wherein the instructions to calculate the modelling error for the second power law function comprises instructions to calculate a total square error between the transfer function and the second power law function.

31. (Previously Presented) The computer program product of claim 18, wherein the instructions to calculate the modelling error for the second power law function comprises instructions to calculate the maximum absolute difference between the transfer function and the second power law function.

32. (Previously Presented) The computer program product of claim 18, wherein the received transfer function comprises a plurality of transfer function values.

33. (Previously Presented) The computer program product of claim 18, wherein the received transfer function comprises a piecewise continuous monotonically increasing function.

34. (Currently Amended) A computer program product, implemented on a machine readable storage device, ~~for adjusting the color information of an image by modelling a non-linear transfer function with a power law function,~~ the computer program product comprising instructions operable to cause a programmable processor to:

receive a transfer function, wherein the transfer function specifies a set of output values corresponding to a set of input values;

fit the transfer function with a first power law function; and

to iteratively, until a termination flag is set:

reflect the first power law function about the transfer function to generate an auxiliary function;

~~fit the auxiliary function with a second power law function~~ define a second power law function by fitting a power law function to the auxiliary function;

calculate a modelling error ~~from~~ between the second power law function and the transfer function, wherein the modelling error is an error value obtained from ~~a plurality of~~ local differences between the second power law function and the transfer function;

identify the first power law function with the second power law function for a next iteration when the modelling error is greater than or equal to a predetermined value; and

set the termination flag when the modelling error is less than [[a]] the predetermined value; and

use the second power law function to calculate a gamma value that is used to adjust the color information of the image.